



# Satellite selection with an end-to-end deep learning network

Panpan Huang<sup>1</sup> · Chris Rizos<sup>1</sup> · Craig Roberts<sup>1</sup>

Received: 13 March 2018 / Accepted: 11 August 2018 / Published online: 16 August 2018  
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

## Abstract

Benefiting from multi-constellation Global Navigation Satellite Systems (GNSS), more and more visible satellites can be used to improve user positioning performance. However, due to limited tracking receiver channels and power consumption, and other issues, it may be not possible, or desirable, to use all satellites in view for positioning. The optimal subset is generally selected from all possible satellite combinations to minimize either Geometric Dilution of Precision (GDOP) or weighted GDOP (WGDOP). However, this brute force approach is difficult to implement in real-time applications due to the time- and power-consuming calculation of the DOP values. As an alternative to a brute force satellite selection procedure, the authors propose an end-to-end deep learning network for satellite selection based on the PointNet and VoxelNet networks. The satellite selection is converted to a satellite segmentation problem, with specified input channel for each satellite and two class labels, one for selected satellites and the other for those not selected. The aim of the satellite segmentation is that a fixed number of satellites with the minimum GDOP/WGDOP value can be segmented from any feeding order of input satellites. To validate the proposed satellite segmentation network, training and test data from 220 IGS stations tracking GPS and GLONASS satellites were used. The segmentation performance using different architectures and representations of input channels, including receiver-to-satellite unit vector and elevation and azimuth, were compared. It was found that the input channel with elevation and azimuth can achieve better performance than using the receiver-to-satellite unit vector, and an architecture with stacked feature encoding (FE) layers has better satellite segmentation performance than one without stacked FE layers. In addition, the models with GDOP and WGDOP criteria for selecting 9 and 12 satellites were trained. It was demonstrated that the satellite segmentation network was about 90 times faster than using the brute force approach. Furthermore, all the trained models can effectively select the satellites making the most contribution to the desired GDOP/WGDOP value. Approximately 99% of the tests had GDOP and WGDOP value differences smaller than 0.03 and 0.2, respectively, between the predicted subset and the optimal subset.

**Keywords** Satellite selection · GDOP/WGDOP · GNSS · Deep learning network

## Introduction

With the deployment of the Galileo and BeiDou GNSS constellations and ongoing modernization of the GPS and GLONASS constellations, the number of satellites in view will be increased to well over 40 most of the time, at many places around the world. The significant increase in the number of satellites from multiple GNSS constellations will greatly improve the navigation performance in terms of positioning accuracy, reliability and availability. However, it may

not always be possible to process all visible satellites in real-time for a standalone GNSS receiver, especially in the case of low-cost receivers with limited tracking channels, or insufficient bandwidth for augmentation message channels, or critical power consumption (Walter et al. 2016). Even if all visible satellites are used for positioning, the positioning accuracy may not necessarily be improved, and real-time performance can deteriorate due to the high computational burden (Blanco-Delgado et al. 2017). The challenge is, how to select the best subset of visible satellites?

The most straightforward algorithm for satellite selection is based on a brute force approach that aims to minimize either the Geometric Dilution of Precision (GDOP) or the weighted GDOP (WGDOP) (Zhang and Zhang 2009). The optimal satellite subset is determined by computing GDOP/

✉ Panpan Huang  
p.huang@student.unsw.edu.au

<sup>1</sup> School of Civil and Environmental Engineering, University of New South Wales, Sydney, NSW 2052, Australia

WGDP values with all possible satellite subset combinations and selecting the one with the minimum value. The GDOP/WGDP calculation cycle can reach billions of computations for one epoch, with the calculation of a single GDOP or WGDP value requiring matrix multiplication and inversion operations. This makes the brute force approach difficult to implement for real-time applications. To reduce the number of GDOP/WGDP calculation cycles, a number of sub-optimal satellite selection methods have been proposed. For example, a quasi-optimal subset of satellites is selected by recursively removing the satellite that has the smallest increase in GDOP from all satellites in view (Liu et al. 2009), or by sequentially adding the least redundant satellite with respect to previously selected satellites (Peng et al. 2014; Roongpiboonsopit and Karimi 2009), or just by removing the satellites that have GDOP contribution values smaller than a predefined threshold (Li et al. 2012). However, these algorithms are very likely to select a globally sub-optimal set of satellites. To track the optimal subset of satellites over time, a temporal algorithm for satellite subset selection has been proposed by evolving the best subset over time by swapping one or two satellites (Swaszek et al. 2017). This algorithm shows great promise in finding the optimal subset. However, this algorithm still suffers from the computational burden of hundreds, even thousands of calculation cycles. To avoid GDOP/WGDP calculations alternative satellite selection measures, such as maximization of the volume of the polytope formed by the satellites (Blanco-Delgado and Nunes 2010; Kong et al. 2014) and satellites' comparability and distribution characteristics (Li et al. 2016; Wei et al. 2012), have been proposed. These methods cannot guarantee optimal satellite selection and need to take geometric satellite distribution into account. Furthermore, instead of directly computing GDOP using matrix multiplication and inversion operations, there are alternative methods. These include closed-form formulas (Doong 2009; Teng and Wang 2016) and machine learning (ML) based methods, such as the Genetic Algorithm (GA) (Mosavi 2011; Zhu 2018), Support Vector Machine (SVM) (Wu et al. 2011), and Neural Network (NN) approaches (Simon and El-Sherief 1995; Jwo and Lai 2007; Azami and Sanei 2014; Zarei 2014). All these ML-based methods treat satellite selection as a regression problem of GDOP calculation and focus on improving the performance of GDOP approximation and/or classification. For example, Wu et al. (2011) compared the performance of several ML methods with different types of training patterns that describe the input–output relationships for GDOP. They found that SVM gives the best performance. Furthermore, different training patterns also affect the performance of the GDOP approximation. However, only around 65% of GDOP value difference smaller than 0.5 can be achieved. Besides, one limitation of these methods is their application locality, which has

to be near the trained location. The performance of some of these methods also depends on the training time and size of training data. To reduce the training time of back propagation (BP) NN, there have been many methods proposed in using either different NNs, such as the probabilistic NN and the general regression NN (Jwo and Lai 2007), or improved BP algorithms, including resilient BP and conjugate gradient algorithms (Azami et al. 2013). Even with the GDOP approximation/classification methods, satellite selection still requires a brute force procedure to identify the subset with the smallest GDOP value from all possible subsets. Moreover, the GDOP classification can only approximately classify the GDOP value of each satellite subset into a predefined range (Jwo and Lai 2007; Azami et al. 2013).

The authors propose an end-to-end deep learning network to select the optimal subset from the set of all visible satellites. Instead of treating the satellite selection as a regression problem for GDOP calculation, it can be considered as a segmentation problem of a small point cloud (Grilli et al. 2017), by partitioning all visible satellites into two classes indicating whether the satellite is selected or not. There is no need to design a specific training pattern to describe the input–output relationships for the GDOP and to implement a brute force selection procedure. To design such an end-to-end deep learning network for satellite selection with GDOP/WGDP criteria, the interactions among all the input satellites need to be captured by the network. Since different satellite feeding orders would not change the satellite segmentation results the network can be designed to be invariant to all permutations of the input, as inspired by PointNet via using a simple symmetric function, max pooling (Qi et al. 2016). Furthermore, the max pooling enables the network to learn the global features of the satellite input. To output the per-satellite segmentation, the local features of each satellite also have to be learned. The PointNet has shown to be effective in per point segmentation by combining local point and global input features learned from multi-layer perceptron (MLP) on each point and max pooling across points, respectively. Therefore, the module with local and global information combination in the PointNet is also adopted by the end-to-end deep learning network for satellite selection. To learn the features of the selected subset from all satellites in view, the input channels used for each satellite should be able to characterize its specified features to help satellite segmentation with respect to GDOP/WGDP criteria. They can be represented by the receiver-to-satellite vector or elevation and azimuth angles. Each visible satellite is processed identically and independently at the beginning using a few fully connected (FC) layers on each satellite to achieve the local satellite features. Then, new per-satellite features are obtained by concatenating the local satellite feature and global satellite input feature with max pooling, which are used as input for the subsequent segmentation layers.

To learn more complex features for the selected satellites with GDOP/WGDOP criteria, the architecture of stacked voxel feature encoding (VFE) layers used in VoxelNet is also employed (Zhou and Tuzel 2017). However, to obtain better trained models for satellite segmentation, the stacked VFE architecture is modified with reduced output sizes to achieve compact satellite internal representations. One problem with this network is that the number in the predicted subset may not always be equal the required one. Since the output scores of the network represent probabilities of input being the predicted labels, this problem can be solved by selecting the required number of satellites according to their output scores.

Following an introduction of the architectures of PointNet and VoxelNet, the end-to-end network for satellite segmentation is described in detail, including the training and test data generation, architecture design, and training details. Models for GDOP and WGDOP with different numbers of satellites are then tested with GNSS observations from 220 IGS stations. Finally, some concluding remarks are given.

### PointNet and VoxelNet networks

PointNet is a unified architecture for tasks with irregular input format such as point cloud-based 3D classification and segmentation (Qi et al. 2016). It is able to directly take point sets as input and outputs either class labels for the entire input or per point labels for each point of the input. To realize satellite selection, each satellite has to be identified to be selected or not. Since the segmentation network of PointNet is capable of partitioning input point sets into multiple homogeneous segments, i.e., per point segment labels for each point of the input, this network is of more interest. Each point in the point cloud can be represented by its  $(x, y, z)$  coordinates as well as other feature channels. Assuming that there are  $n$  unordered input points  $\{x_1, x_2, \dots, x_n\}$  with  $x_i \in \mathbb{R}^d$  and  $m$  sub-categories, the model trained from the segmentation network is able to output  $n \times m$  scores for each point and sub-category. The basic idea of the PointNet is to approximate a continuous function  $f : \mathcal{X} \rightarrow \mathbb{R}$  that maps the unordered input points to a vector by applying a symmetric function, max pooling, on transformed elements in the set:

$$f(\{x_1, x_2, \dots, x_n\}) \approx \gamma\left(\text{MAX}_{i=1,2,\dots,n} \{p(x_i)\}\right) \tag{1}$$

where  $\gamma$  and  $p$  are usually multi-layer perceptron (MLP) networks. The output vector  $\mathbf{f}$  represents the global signature of the input set. Each input point is identically and independently processed at the initial stages using the MLP with different hidden layers to encode statistical properties of the points. The MLP can also be seen as several FC layers

operating on each point, i.e.,  $1 \times 1$  convolution layers. To achieve per point segmentation, both local and global features of the input point are extracted to learn the interactions among the points. The local features of each point  $\mathbf{p}_i \in \mathbb{R}^l$  can be obtained from the initial stages with the MLP, and the global features  $\mathbf{f} \in \mathbb{R}^g$  can be achieved with the max pooling algorithm across points. Then the new per point features  $\mathbf{p}_i^{\text{new}} \in \mathbb{R}^{l+g}$  are extracted by concatenating the local point features with the global features:

$$\mathbf{p}_i^{\text{new}} = [\mathbf{p}_i^T \ \mathbf{f}^T]^T \tag{2}$$

Finally, the combined new per point features are used for point label prediction with the MLP. Figure 1 illustrates the basic segmentation network architecture.

VoxelNet is an end-to-end network for point cloud-based 3D object detection (Zhou and Tuzel 2017). The voxel used in the VoxelNet refers to a 3D grid, which contains a number of points divided from the point cloud input. One key innovation of the VoxelNet is its architecture of stacked VFE layers to learn voxel-wise feature, as shown in Fig. 2. As with the feature aggregation network in PointNet, each VFE layer is able to achieve the new point features by combining the point-wise features with one FC layer and the locally aggregated features with the max pooling, i.e., the local point features and global voxel features. The main difference between the VoxelNet feature learning architecture and PointNet is that there are several global and local feature concatenations performed with the stacked VFE layers. To illustrate the detailed processing procedure of one voxel, all the feature sizes involved in the different layers are shown in Fig. 2. It is assumed that the network also takes  $n$  points with input channels  $d$  as input and outputs voxel feature  $f_v \in \mathbb{R}^v$ . Since there is one FC layer and one max pooling in each VFE layer, the output feature size from each VFE is two times the local feature size. The stacked VFE layers also directly consume the points within the voxel as input and outputs new per point concatenated features. It encodes point interactions within

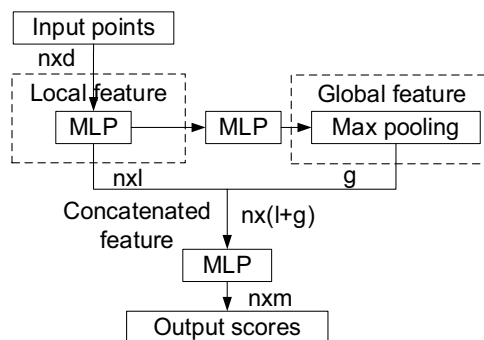


Fig. 1 PointNet segmentation network architecture

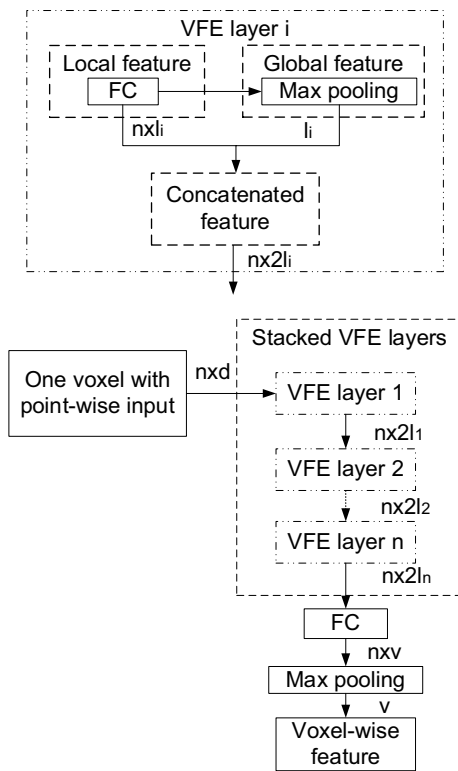


Fig. 2 Architecture of stacked VFE layers

a voxel and enables learning complex voxel-wise features for characterizing 3D shape information of each voxel.

### An end-to-end satellite segmentation network

In contrast to the GDOP approximation/classification methods mentioned in the introduction that treat satellite selection as a regression problem and classifying the GDOP value of satellite input into a predefined range, the proposed end-to-end satellite segmentation network is able to directly segment all the input satellites selected or not with no need for the brute force procedure as required in GDOP approximation/classification methods. One key problem for satellite segmentation using GDOP/WGDOP criteria is that the segmentation results should be invariant to all the permutations of the input satellites. Inspired by the PointNet segmentation network that uses max pooling to deal with unordered input, and predicts the per point class label by combining the global feature with local point features, this architecture can be used for satellite segmentation network by treating the satellites in view at one instant as point cloud input, with two class labels—for selecting the satellite or not selecting the satellite—as output. In addition, the architecture of stacked VFE layers in the VoxelNet is employed

to improve the performance of satellite segmentation using GDOP/WGDOP criteria.

### Train and test data generation

Assume the selection of  $m$  satellites from  $n$  satellites in view from multiple constellations where inter-system time offsets have been accounted for. The criterion used for satellite selection is to minimize either the GDOP or WGDOP value. These values can be calculated by:

$$GDOP = \sqrt{\text{tr}(\mathbf{H}^T \mathbf{H})^{-1}} \quad WGDOP = \sqrt{\text{tr}(\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1}} \quad (3)$$

where  $\text{tr}(\cdot)$  denotes trace of the matrix,  $\mathbf{H}$  is the design matrix and  $\mathbf{W}$  is the weight matrix.  $\mathbf{H}$  and  $\mathbf{W}$  can be represented by:

$$\mathbf{H} = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} w_1 & & & \mathbf{0} \\ & w_2 & & \\ & & \ddots & \\ \mathbf{0} & & & w_n \end{bmatrix} \quad (4)$$

where  $h_i = [x_i \ y_i \ z_i]$  is the receiver-to- $i$ th-satellite unit vector, the column of ones in the  $\mathbf{H}$  matrix represents parameters for the receiver clock bias in units of meters and  $w_i$  is the weight of the  $i$ th satellite. Using the brute force approach, the subset with the minimum GDOP or WGDOP value can be selected from  $n!/[m!(n-m)!]$  possible combinations. Therefore, there are two classes in the output data. One class is for the  $m$  selected satellites and the other is for the  $n - m$  not-selected satellites. To classify the  $n$  satellites in an end-to-end fashion, the input data of each satellite should be able to characterize its local feature. The receiver-to-satellite unit vector  $[x_i \ y_i \ z_i]$  can be used as the input channel, which contains direct information for the optimal subset selection using GDOP/WGDOP criteria. Since it can also be converted to elevation and azimuth  $[el_i \ az_i]$ , the input channel represented by  $[el_i \ az_i]$  is another option. In the following experiment, the satellite segmentation performance using both forms of input channels is compared.

### Satellite segmentation network architecture

To segment all the satellites in view, both local and global features of each satellite are needed. The local feature of each satellite can be obtained by applying inputs and feature transformation through a few FC layers on each satellite. Since the satellite order does not change the final segmentation result, max pooling is used for the satellite segmentation network. Furthermore, max pooling also learns the global features of the satellite input. Then new satellite-wise features can be extracted by combining the global features with the local satellite features. To learn multi-level features for characterizing the pattern of the selected satellites, stacked

feature encoding (FE) layers based on the architecture of stacked VFE layers are used. However, instead of concatenating local and global features every time the local feature is obtained with the FC layer in each VFE layer, the MLP with a few FC layers are used. As with the MLP in the PointNet, the FC layers operate on each satellite. The global feature obtained from the last layer in the MLP is concatenated with the local feature in the same layer or preceding one as

shown in Fig. 3. All the feature sizes derived from different layer are also illustrated in Fig. 3. This architecture is able to reduce the inference time due to the two times increase in feature size resulting from feature concatenation of each FC layer as in the stacked VFE layers.

### Training details

Since the number of satellites in view will change with time, zero-padding is needed. The size of the zero-padding can be determined by fixing the number of observed satellites to the maximum possible number of visible satellites, e.g.,  $n$ . Therefore, the size of each input training data is  $n \times d$ , where  $d$  is the dimension of the satellite input channel. All training data are randomized before processing. The network is designed by trading off the computational complexity and performance of satellite segmentation. It is composed of seven stacked FE layers and two FC layers for the final segmentation. In each FE layer, there are three FC layers and a max pooling. The max pooling, performed across the local features learned from the last FC layer, is used to obtain the global feature. The global feature is then concatenated with the local feature obtained from the second FC layer, as shown in Fig. 3. Overall, there are 23 layers in the proposed network architecture. The detailed layer output sizes are listed in Table 1. The number of layers and their sizes are empirical values.

The gradual increase in the channel size makes it possible for the network to learn more high-level features. Due to the redundancy in the concatenated feature in each FE layer, the channel size of the FC layer in the next FE layer is reduced to keep the compact satellite features. All FC layers use rectified linear units (ReLU) as activation function and batch normalization. Since the satellite segmentation is satellite-wise classification, all of the input data including both satellite-input and zero-padding input have to be segmented. The same label as satellite not-selected is assigned to zero-padding input. The loss function for the

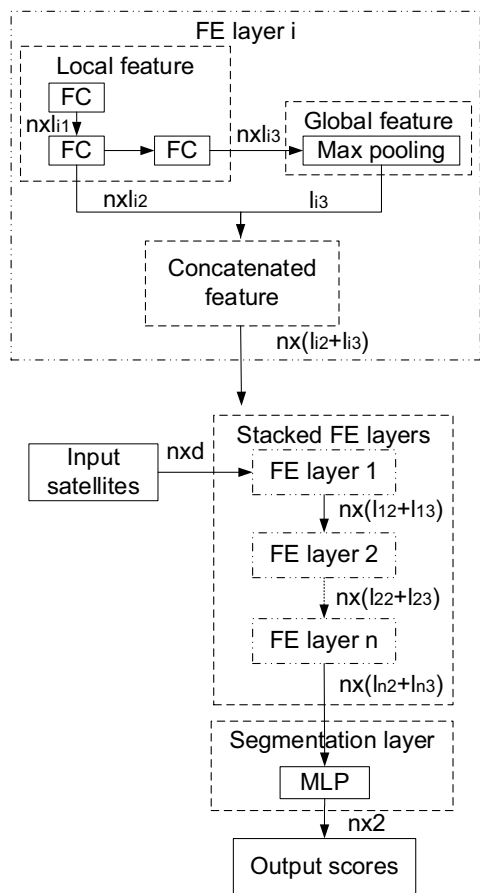


Fig. 3 Satellite segmentation network architecture

Table 1 Satellite segmentation architecture with stacked FE layers

Layers		Output size ( $l_{i1}, l_{i2}, l_{i3}$ )	Concatenated feature size ( $l_{i2} + l_{i3}$ )	Number of FC layers
Stacked FE layers	FE layer 1	(32, 32, 48)	80	3
	FE layer 2	(64, 64, 96)	160	3
	FE layer 3	(128, 128, 160)	298	3
	FE layer 4	(256, 256, 320)	576	3
	FE layer 5	(512, 512, 640)	1152	3
	FE layer 6	(864, 864, 960)	1824	3
	FE layer 7	(1024, 1024, 1152)	2176	3
Segmentation layer		(1024, 2)	—	2



satellite segmentation network is based on the binary cross entropy between the target and predicted classes (De Boer et al. 2005):

$$L = -\alpha \frac{1}{N_1} \sum_{i=1}^{N_1} \log(\hat{l}^i) - \beta \frac{1}{N_0} \sum_{i=1}^{N_0} \log(1 - \hat{l}^i) - \gamma \frac{1}{N_c} \sum_{i=1}^{N_c} \log(1 - \hat{l}^i) \tag{5}$$

where  $N_1$ ,  $N_0$ , and  $N_c$  denote the number of satellite-input training data with label 1, 0 and zero-padding input data with label 0, respectively;  $\hat{l}^i$  denotes the predicted value of the softmax output; and  $\alpha$ ,  $\beta$ , and  $\gamma$  are loss weight constants to adjust the relative importance balance among different input data. The minimization of this loss function also indirectly minimizes the difference in the GDOP or WGDOP values between the predicted satellites to be selected and the targeted ones. This loss function is optimized based on one commonly used stochastic gradient descent algorithm for deep learning networks, the Adam optimization algorithm (Kingma and Ba 2014). This algorithm uses first-order gradients to optimize stochastic objective functions with adaptive estimates of first and second moments. In addition, the algorithm has very little memory requirement and is very suitable for neural networks with large training datasets and parameters. All the computations in the proposed network can be implemented on a CPU/GPU in parallel. One problem when using this network is that the number of output predictions for selected satellites may not be equal to the predefined one. As the output scores of the network can be interpreted as probabilities of input being the predicted labels, the number of output satellites for selection can be fixed by selecting the satellites with high output scores for label 1.

### Experiment

Based on the brute force approach, training data with GDOP and WGDOP criteria were generated using 1-day GNSS observations (at 1-min intervals) from 200 IGS stations, resulting in around 288,000 training data samples in total. Both GPS and GLONASS observations were used and it was assumed that the inter-frequency bias for GLONASS satellites had been accounted for. Elevation cutoff angle for each receiver was set to 15°. Test data were generated in a similar way for another 20 IGS stations. The maximum number of satellites in view was 20, i.e.,  $n = 20$ . The labels for selected satellites were set as class 1 and the not-selected satellites were designated class 0. Due to the slightly unbalanced training data, different loss weight values  $\alpha = 1$ ,  $\beta = 1.2$ , and  $\gamma = 0.05$  were assigned. Adam optimizer with initial learning rate 0.1, momentum 0.9 and batch size 128 values were used. The decay rate for batch normalization was 0.7, starting from 0.5 and gradually increasing to 0.99. The learning rate was reduced at the same rate as the decay

rate for batch normalization. The models were converged in around 50 min using TensorFlow (Abadi et al. 2016), an open-source machine learning library running on a GTX1080TI GPU. The code and trained models used in the following experiment can be found at [https://github.com/PanUnsw/satellite\\_selection.git](https://github.com/PanUnsw/satellite_selection.git).

### Satellite segmentation with different input channels and architectures

To compare the segmentation performance with input channels represented by  $c_i = [x_i \ y_i \ z_i]$  and  $c_i = [el_i \ az_i]$ , one example for selecting 9 satellites based on the GDOP criterion is shown in Fig. 4. Table 2 is a comparison of training and testing accuracies. The accuracy represents the percentage of satellite input segmented correctly, i.e., the predicted subset is the same as the optimal subset. Although  $[x_i \ y_i \ z_i]$  and  $[el_i \ az_i]$  are mathematically equivalent, the input channel with  $[x_i \ y_i \ z_i]$  has slower training convergence, and worse converged training and testing accuracies than that based on  $[el_i \ az_i]$ . This fact can be intuitively explained by

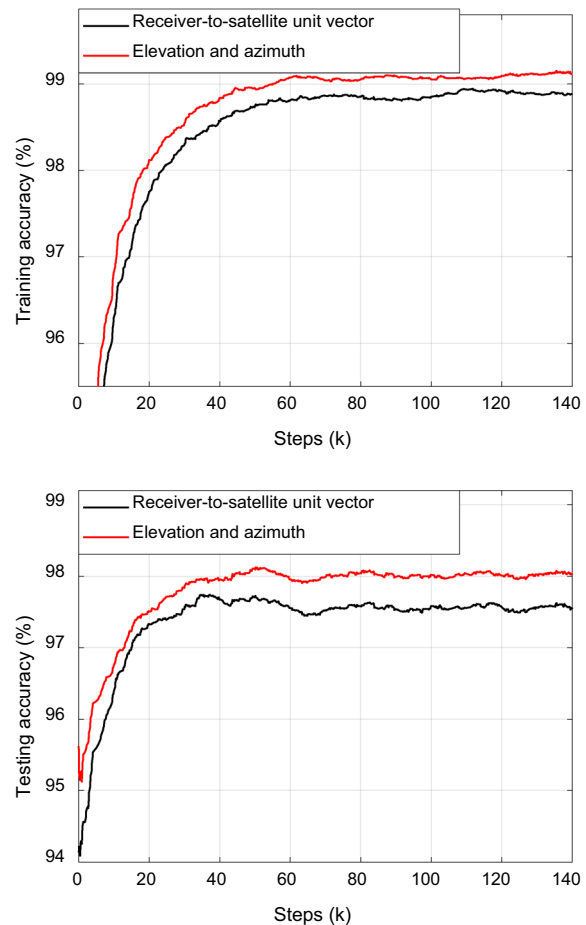


Fig. 4 Performance comparison with different input channels

**Table 2** Accuracy comparison with different input channels

Input channels	Training accuracy (%)	Testing accuracy (%)
Receiver-to-satellite unit vector	98.9	97.6
Elevation and azimuth	99.1	98.0

**Table 3** Satellite segmentation architecture without stacked FE layers

Layers	Output size
FC layers	(32, 64, 128, 256, 512, 1024)
Max pooling	1024
Segmentation layer	(512, 2)

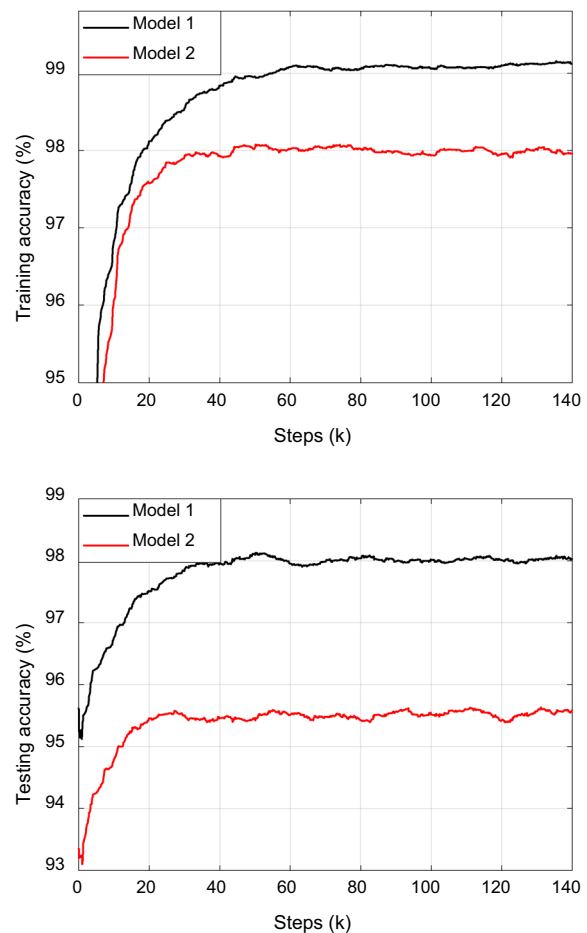
**Table 4** Accuracy comparison with different architectures

Network architecture	Training accuracy (%)	Testing accuracy (%)
Model 1	99.1	98.0
Model 2	97.9	95.6

the saddle points when optimizing the loss function, which can considerably slow down training (Dauphin et al. 2014). It may take more time to escape the saddle points for the input channel with  $[x_i y_i z_i]$  compared with  $[el_i az_i]$  at the beginning of training. The input channels represented by  $[el_i az_i]$  were used in the following experiment.

To illustrate the performance of the satellite segmentation architecture with stacked FE layers compared with that without stacked FE layers, another model with an architecture without stacked FE layers was also trained with elevation and azimuth input channels for selecting nine satellites based on the GDOP criterion. The architecture, consisting of six FC layers, one max pooling layer and one segmentation layer was used by trading off the computational complexity and performance of satellite segmentation. The output sizes of these layers are shown in Table 3. Local features obtained from the third FC layer were concatenated with the global features from the max pooling layer. The number of layers and their sizes used for the architecture without stacked FE layers are also empirical values (Table 3).

The initial parameters used for the architecture without stacked FE layers were the same as with stacked FE layers except that the decay rates for batch normalization and learning rate were set to 0.3 to reduce the tendency for overfitting. Figure 5 and Table 4 show the performance comparison



**Fig. 5** Performance comparison with different architectures

between the two trained models, model 1 and 2 with and without stacked FE architecture, respectively. It can be seen that both the training and test accuracies of model 2 are worse than that of model 1. Therefore, it is preferable to use the architecture with stacked FE layers for satellite segmentation.

**Satellite segmentation with GDOP**

Models for selecting 9 and 12 satellites from the same set of observed satellites using the GDOP criterion were trained and evaluated. With the same network architecture with stacked FE layers, the models trained were used when selecting, in turn, 9 or 12 satellites. Table 5 shows the segmentation performance comparison. It can be seen that the testing accuracy is slightly worse than the training accuracy. The model for selecting 12 satellites has better performance than that for 9 satellites.

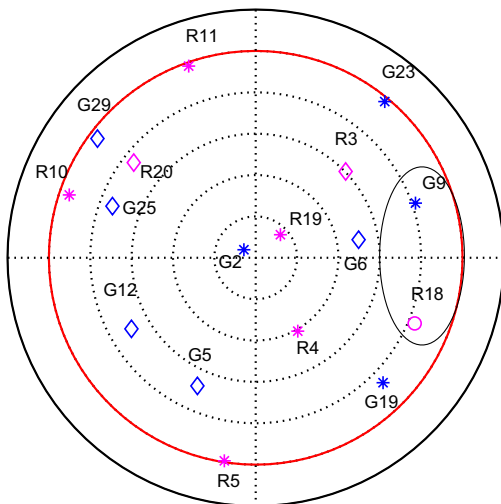
To validate the trained models, Table 6 lists the test results. Compared with means of the best GDOP values, the means of the test GDOP values were only increased

**Table 5** Satellite segmentation performance using the GDOP criterion

Number of selected satellites	Training accuracy (%)	Testing accuracy (%)
9	99.1	98.0
12	99.2	98.3

**Table 6** Test evaluation performance with GDOP-trained model

Number of selected satellites	Mean of best GDOP	Mean of test GDOP	Percentage of wrong segmentation	
			0 (%)	1 (%)
9	1.981	1.983	81	18
12	1.809	1.810	83	16



**Fig. 6** One example of a wrongly predicted subset with GDOP-trained model. Asterisk, diamond and circle markers identify the satellites optimally selected, not selected and wrongly selected, respectively

by 0.08 and 0.04% when selecting 9 or 12 satellites, respectively. For more than 80% of the epochs the optimal subset of the satellite input was correctly predicted, and around 20% when one satellite in the optimal subset was not selected. The one satellite in the optimal subset that was not selected was often replaced by one that produces a slightly worse GDOP value than the minimum GDOP value.

One example of the predicted subset with one wrongly selected satellite is shown in Fig. 6. The detailed input and output labels for the targets and predictions are listed in Table 7. The satellite PRN G9 in the optimal subset was predicted with label 0, which was replaced by satellite PRN

**Table 7** One example of satellite segmentation with GDOP-trained model

Satellite	Input		Output labels	
	Elevation (rad)	Azimuth (rad)	Targets	Predictions
G2	1.480	5.310	1	1
G5	0.680	3.569	0	0
G6	0.909	1.397	0	0
G9	0.502	1.241	1	0
G12	0.664	4.193	0	0
G19	0.442	2.346	1	1
G23	0.288	0.690	1	1
G25	0.607	5.058	0	0
G29	0.317	5.358	0	0
R3	0.782	0.806	0	0
R4	1.035	2.621	1	1
R5	0.270	3.295	1	1
R10	0.325	5.037	1	1
R11	0.285	5.946	1	1
R18	0.482	1.962	0	1
R19	1.356	0.820	1	1
R20	0.591	5.375	0	0

R18 resulting in the optimal GDOP value increasing from 1.617 to 1.619.

Figure 7 is a plot of the percentage of GDOP increase between the predicted and the optimal subsets. It can be seen that the GDOP increase is limited to 8 and 3% when selecting 9 or 12 satellites, respectively, as shown in the top and bottom panels.

The histogram of GDOP value differences between the test and the optimal GDOP values is plotted in Fig. 8. For about 99 and 100% of the time, the GDOP difference is smaller than 0.03 when selecting 9 or 12 satellites, respectively.

### Satellite segmentation with WGDOP

In this section, satellite segmentation results are presented for two trained models using the WGDOP criterion. The weight calculated for each satellite is based on its elevation angle as  $w = [\sin(e)]^2$ . With the same network architecture used above, the best performance results of two trained models when selecting 9 or 12 satellites are listed in Table 8. Similar to the earlier trained models using the GDOP criterion, the model for selecting 12 satellites has better performance than that for 9 satellite selection. However, both models trained with WGDOP have lower accuracies than the ones with GDOP. This can be attributed to more complex features introduced by the different target satellites.

Even with the larger bias between the testing and training accuracies, around 95% of the time the predicted subsets



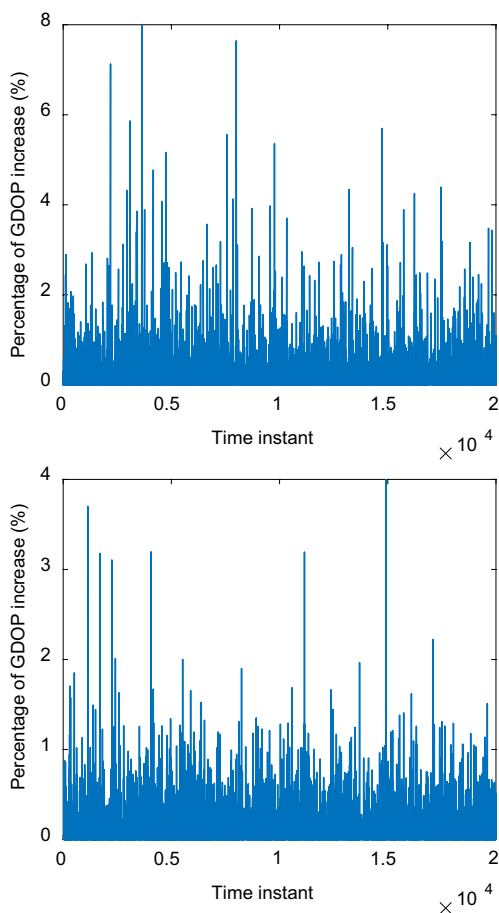


Fig. 7 Percentage of GDOP increase

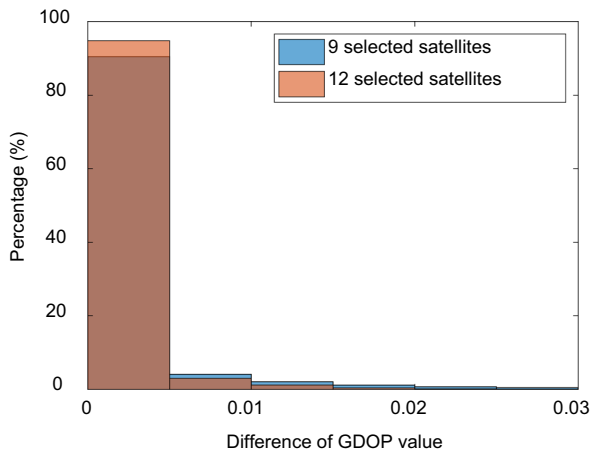


Fig. 8 GDOP value difference comparison

are correctly selected, or with just one wrong segmentation, compared with the optimal ones, as listed in Table 9. The means of the test WGDOP values are increased by 0.31 and 0.15% when selecting 9 or 12 satellites, respectively. Similar to the earlier case, the subset with one wrongly selected

Table 8 Satellite segmentation performance with the WGDOP criterion

Number of selected satellites	Training accuracy (%)	Testing accuracy (%)
9	99.1	95.8
12	99.2	96.8

Table 9 Test evaluation performance with WGDOP-trained model

Number of selected satellites	Mean of best WGDOP	Mean of test WGDOP	Percentage of wrong segmentation	
			0 (%)	1 (%)
9	3.742	3.753	65	29
12	3.355	3.359	71	27

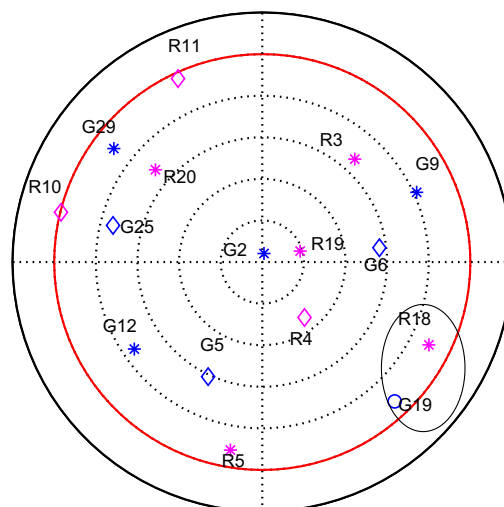


Fig. 9 One example of a wrongly predicted subset with WGDOP-trained model. Asterisk, diamond and circle markers identify the satellites optimally selected, not selected and wrongly selected, respectively

satellite usually has a slightly larger WGDOP value than the optimal subset. One typical example of a wrongly predicted subset is shown in Fig. 9. Table 10 lists the elevation and azimuth input and the output labels for the targets and predictions. The difference in WGDOP value between the optimal and predicted subsets is approximate 0.012, with WGDOP increased from 3.418 to 3.430.

The percentage of WGDOP increase between the predicted and the optimal subsets is mostly limited to 8 and 4% when selecting 9 or 12 satellites, respectively, as shown in the top and bottom panels of Fig. 10.

Figure 11 shows the histogram of WGDOP value differences between the test and the optimal WGDOP values.

**Table 10** One example of satellite segmentation with WGDOP-trained model

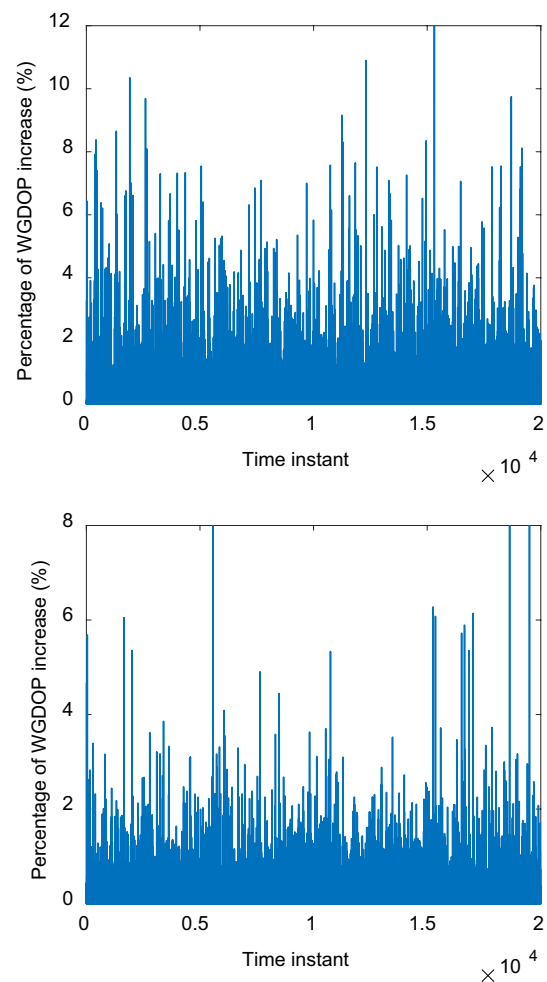
Satellite	Input		Output labels	
	Elevation (rad)	Azimuth (rad)	Targets	Predictions
G2	1.516	0.197	1	1
G5	0.773	3.582	0	0
G6	0.828	1.449	0	0
G9	0.505	1.146	1	1
G12	0.597	4.114	1	1
G19	0.361	2.382	0	1
G25	0.603	4.953	0	0
G29	0.395	5.365	1	1
R3	0.699	0.731	1	1
R4	1.132	2.491	0	0
R5	0.370	3.309	1	1
R10	0.266	4.955	0	0
R11	0.300	5.853	0	0
R18	0.400	2.032	1	0
R19	1.322	1.289	1	1
R20	0.682	5.425	1	1

More than 99% of the differences in WGDOP value are less than 0.2, for both cases of selecting 9 or 12 satellites. This is adequate for most applications.

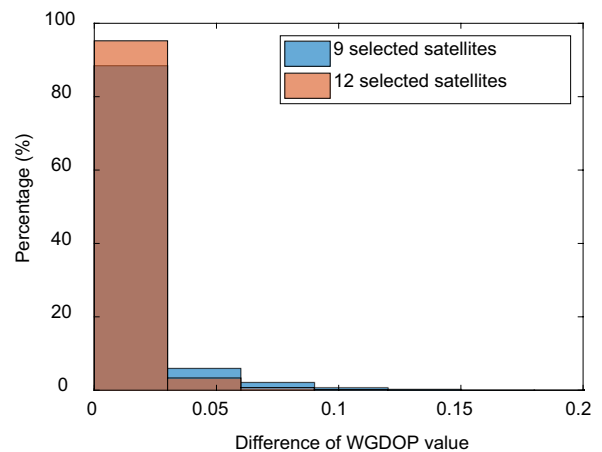
The comparison of average computational time with GDOP and WGDOP criteria between the brute force approach to select one optimal subset and the proposed satellite segmentation network to predict one selected subset with the trained models is summarized in Table 11. It can be seen that the satellite segmentation network method is about 90 times faster than the brute force approach.

### Concluding remarks

The authors presented an end-to-end deep learning network for satellite selection invariant to satellite input permutation based on the PointNet and VoxelNet networks. The satellite selection procedure was converted to a satellite segmentation procedure, with specified input channel for each satellite and two class labels representing the selected and not-selected satellites. The proposed satellite segmentation network was composed of several simple stacked FE layers and one segmentation layer. An experiment was conducted to evaluate the proposed approach with training and test data from 220 IGS stations. The satellite segmentation performance was compared with respect to different input channels, including receiver-to-satellite unit vector and elevation and azimuth, as well as different architectures, i.e., with and without stacked FE layers. The experiment showed that it was preferable to use an architecture with stacked FE layers and



**Fig. 10** Percentage of WGDOP increase



**Fig. 11** WGDOP value comparison

input channel represented by elevation and azimuth due to the faster training convergence and better converged accuracy. Cases for selecting 9 or 12 satellites, with GDOP and

**Table 11** Performance comparison of computation time

Methods	Average computation time (ms)	
	GDOP	WGDOP
Brute force approach	110	115
Satellite segmentation network	1.2	1.3

WGDOP criteria, were investigated. It was demonstrated that using the satellite segmentation network approach was around 90 times faster than the brute force satellite selection approach. From the test results, it can be concluded that the trained models were capable of selecting the satellites that had the most contribution to the GDOP or WGDOP value and had no limitation of application locality as shown in GDOP approximation/classification methods. In addition, the trained models based on GDOP had better performance than the ones based on WGDOP. Furthermore, the models for selecting 12 satellites were more accurate than those for nine satellites. The model proposed for satellite segmentation is only for selecting a single fixed number of satellites, and the training and test data are all from static receivers. In the future, a model capable of selecting a multiple number of satellites will be trained with the selected number of satellites added to the input channel. Kinematic data will be analyzed, and the robustness of the trained model will be investigated.

**Acknowledgements** This research is supported by the Chinese Scholarship Council (CSC) awarded to the Panpan Huang.

## References

- Abadi M et al (2016) Tensorflow: large-scale machine learning on heterogeneous distributed systems (arXiv preprint: arXiv:1603.04467)
- Azami H, Sanei S (2014) GPS GDOP classification via improved neural network trainings and principal component analysis. *Int J Electron* 101(9):1300–1313
- Azami H, Mosavi MR, Sanei S (2013) Classification of GPS satellites using improved back propagation training algorithms. *Wirel Pers Commun* 71(2):789–803
- Blanco-Delgado N, Nunes FD (2010) Satellite selection method for multi-constellation GNSS using convex geometry. *IEEE Trans Veh Technol* 59(9):4289–4297
- Blanco-Delgado N, Nunes FD, Seco-Granados G (2017) On the relation between GDOP and the volume described by the user-to-satellite unit vectors for GNSS positioning. *GPS Solut* 21(3):1139–1147
- Dauphin Y, Pascanu R, Gulcehre C, Cho K, Ganguli S, Bengio Y (2014) Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In: *Advances in neural information processing systems*, Montreal, 08–13 Dec 2014, vol 2 pp 2933–2941
- De Boer PT, Kroese DP, Mannor S, Rubinstein RY (2005) A tutorial on the cross-entropy method. *Ann Oper Res* 134(1):19–67
- Doong SH (2009) A closed-form formula for GPS GDOP computation. *GPS Solut* 13(3):183–190
- Grilli E, Menna F, Remondino F (2017) A review of point clouds segmentation and classification algorithms. *Int Arch Photogramm Remote Sens Spat Inf Sci* 42(2):W3
- Jwo DJ, Lai CC (2007) Neural network-based GPS GDOP approximation and classification. *GPS Solut* 11(1):51–60
- Kingma DP, Ba J (2014) Adam: a method for stochastic optimization (arXiv preprint: arXiv:1412.6980)
- Kong J, Mao X, Li S (2014) BDS/GPS satellite selection algorithm based on polyhedron volumetric method. In: *2014 IEEE/SICE international symposium on system integration*, Tokyo, 13–15 Dec 2014, pp 340–345
- Li G, Xu C, Zhang P, Hu C (2012) A modified satellite selection algorithm based on satellite contribution for GDOP in GNSS. In: *Advances in mechanical and electronic engineering*, volume 1, pp 415–421
- Li G, Wu J, Liu W, Zhao C (2016) A new approach of satellite selection for multi-constellation integrated navigation system. In: *China satellite navigation conference (CSNC) 2016 proceedings*, volume III, pp 359–371
- Liu M, Fortin M-A, Landry R (2009) A recursive quasi-optimal fast satellite selection method for GNSS receivers. In: *Proceedings of the ION GNSS 2009*, Institute of Navigation, Savannah, 22–25 September 2009, pp 2061–2071
- Mosavi M (2011) Applying genetic algorithm to fast and precise selection of GPS satellites. *Asian J Appl Sci Eng* 4(3):229–237
- Peng A, Ou G, Li G (2014) Fast satellite selection method for multi-constellation Global Navigation Satellite System under obstacle environments. *IET Radar Sonar Navig* 8(9):1051–1058
- Qi CR, Su H, Mo K, Guibas LJ (2016) PointNet: deep learning on point sets for 3D classification and segmentation (arXiv preprint arXiv:161200593)
- Roongpiboonsopit D, Karimi HA (2009) A multi-constellations satellite selection algorithm for integrated global navigation satellite systems. *J Intell Transp Syst* 13(3):127–141
- Simon D, El-Sherief H (1995) Navigation satellite selection using neural networks. *Neurocomputing* 7(3):247–258
- Swaszek PF, Hartnett RJ, Seals KC, Swaszek R (2017) A temporal algorithm for satellite subset selection in multi-constellation GNSS. In: *Proceedings of the ION ITM 2017*, Institute of Navigation, Monterey, 30 Jan–2 Feb 2017, pp 1147–1159
- Teng Y, Wang J (2016) A closed-form formula to calculate geometric dilution of precision (GDOP) for multi-GNSS constellations. *GPS Solut* 20(3):331–339
- Walter T, Blanch J, Kropp V (2016) Satellite selection for multi-constellation SBAS. In: *Proceedings of the ION GNSS 2016*, Institute of Navigation, Portland, 12–16 September 2016, pp 1350–1359
- Wei M, Wang J, Li J (2012) A new satellite selection algorithm for real-time application. In: *2012 international conference on systems and informatics (ICSAI2012)*, Yantai, pp 2567–2570
- Wu CH, Su WH, Ho YW (2011) A study on GPS GDOP approximation using support-vector machines. *IEEE Trans Instrum Meas* 60(1):137–145
- Zarei N (2014) Artificial intelligence approaches for GPS GDOP classification. *Int J Comput Appl* 96(16):16–21
- Zhang M, Zhang J (2009) A fast satellite selection algorithm: beyond four satellites. *IEEE J Sel Top Signal Process* 3(5):740–747
- Zhou Y, Tuzel O (2017) VoxelNet: end-to-end learning for point cloud based 3D object detection (arXiv preprint: arXiv:171106396)
- Zhu S (2018) An optimal satellite selection model of global navigation satellite system based on genetic algorithm. In: *China satellite navigation conference (CSNC) 2018 proceedings*



**Panpan Huang** is currently working toward the Ph.D. degree at UNSW. Her research interests include GNSS augmentation system and high precision multi-GNSS/PPP positioning.



**Craig Roberts** is a Senior Lecturer in the School of Civil and Environmental Engineering at UNSW with interests in the application of high precision multi-GNSS/PPP positioning, and implications of 3D kinematic datum modernization for professional and mass market users.



**Chris Rizos** is an Emeritus Professor in the School of Civil and Environmental Engineering at UNSW. He is a fellow of the IAG, a fellow and current President of the Australian Institute of Navigation, a fellow of the U.S. Institute of Navigation, and an Honorary Professor of Wuhan University, Wuhan, China. His research interests include the technology and applications of GPS, GNSS, and other navigation/positioning systems.